SZ12-1012-6

SLiKSFTP

Industrial Grade Data Transfer for z/OS platforms

USER'S GUIDE

Version 1 Release 1

March 2014

Copyright Australian Systems Engineering Pty. Ltd. 2014

LICENSED MATERIAL
COPYRIGHT Australian Systems Engineering Pty. Ltd. 2012,2013,2014
This material always remains the property of Australian Systems Engineering Pty. Ltd. and must be used only in accordance with the terms of the agreement under which it is supplied.

Manual and Product identifiers.

This manual is identified by a publication number, **SZ121012,** combined with a revision level number (sometimes called a dash level) **-6**.

The licensed software product, of which this manual is a part, is identified by a **Product Name** or **Product Acronym** and an **Edition Number**. Edition numbers are made up of a **Version number** and a **Release number** within that version.

The material in this manual is intended to be used with **SLiKSFTP Version 1 Release 1**. The edition number may appear in licensed material in abbreviated forms like the following: **V1R1**, and **V1.1**.

From time to time ASE may issue replacements for, or additions to, the pages in this manual.

Each time the entire manual is replaced by a later version then the revision level (the dash level) part of the publication identifier will be increased by 1.

Publication history:

Dash	product	ver.rel	GA date	Highlights/added features
-4	SLiKSFTP	V1.0	Apr 2013	Initial version
-5	SLiKSFTP	V1.0	Sep 2013	
-6	SLiKSFTP	V1.1	Mar 2014	Support for GET,GETZIP,PUTZIP in FTP mode

Related Publications
None

Published by

Australian Systems Engineering Pty. Ltd.
126A Gillies Street.
FAIRFIELD, Victoria 3078, Australia
telephone +61 3 9419 4200
email: help@slikzip.com

Trademarks

OS/390 and z/OS are registered trademarks of IBM Corp.
PKZIP and PKUNZIP are registered trademarks of PKWARE, Inc.
UNIX is a registered trademark of The Open Group
Windows is a registered trademark of Microsoft Corporation
WinZip is a registered trademark of WinZip Computing, Inc.

Contents

Overview 6
Installing SLiKSFTP 6
Product Activation 7
Installing SLiKSFTP license key 7
Using SLiKSFTP 9
Sample JCL for 9
Running as an FTP client 9
SLiKSFTP Control Statements 9
demokey key 10
ftp 10
open hostname [port] 10
userid username@hostname 11
userid username 11
pass password 11
cd remotedir 11
binary 11
ascii 11
crlf 11
lf 12
ebcdic 12
put mvsdsn|dd:ddname remotefile 12
putzip mvsdsn|dd:ddname remotefile 12
Get remotefile mvsdsn|dd:ddname 12
getzip mvsdsn|dd:ddname remotezipfile 13
chmod mode remotefile 13
chgrp grp remotefile 13
chown owner remotefile 13
exit|bye|quit 13
ls remotefile 13
mkdir dirname 13
rmdir dirname 14
rm remotefile 14
pwd 14
sshconf configfilename 14
sshopt option 14
passive 14
site siteoptions 14
quote string 15
stat 15
syst 15
help 15
options option1[,option2,…] 15
- nostderr 15

- mergeout 15
- replace 15
- noreplace 15
- resume 16
- noresume 16
- progress 16
- noprogress 16
- zipmsgs 16
- nozipmsgs 16
- mow(nnnn) 16
- ignorerr 16
- noignorerr 16
- size(nnnn) 17
- keepalive(nnnn) 17
- retry(nnnn) 17

compopt option1[,option2,…] 17
- ascii 17
- binary 17
- lf 17
- crlf 18
- member(*name*) 18
- opt(*string*) 18
- password(*password*) 18
- replace 18
- noreplace 18
- strip 18
- nostrip 19
- select(*selectionmask*) 19
- text 19
- trunc 19
- wrap 19
- zip64 19

/* (comment-only lines) 20

Sample Jobs 20

1. Transfer file, change file's mode on remote host: 20
2. ZIP & transfer file, change file's mode on remote host: 21
3. ZIP, encrypt & transfer a file, set file's mode on remote host: 21
4. Translate file from EBCDIC to ASCII & transfer it: 22
5. Zip a file, translating from EBCDIC to ASCII, and transfer it. 22
6. Retrieve a file from remote host. 23
7. Retrieve a text file, translating from ASCII to EBCDIC: 23
8. Translate file from EBCDIC to ASCII without removing trailing blanks & transfer it: 24
9. Transfer an EBCDIC text file to another EBCDIC machine: 24

10. ZIP & transfer file, suppress zip messages: 25
11. Transfer a ZIP file from a remote machine and UNZIP: 25
12. Transfer a ZIP file from a remote machine, UNZIP and decrypt: 26
13. ZIP & transfer file using FTP protocol: 26
14. ZIP, encrypt & transfer a file using FTP protocol: 27
15. ZIP selected members from a PDS, translate to ascii, append linefeed to each record and transfer the zip file using FTP protocol: 27
16. Use SLiKZIP control statements to ZIP multiple datasets, translating to ascii , append carriage return/linefeed to each record, remove trailing blanks and transfer the zip file using FTP protocol: 28
17. Transfer a ZIP file from a remote machine and UNZIP a selected member translating from ascii and removing the line feed from the end of each record: 29
18. ZIP & transfer file using an alternate ssh configuration: 29
Sample Output 30
1. The output of a typical SLIKSFTP job using SFTP protocol 30
2. The output of a typical SLIKSFTP job using FTP protocol 31
Messages 33

Overview

SLiKSFTP is utility program that provides direct transfer of z/OS mainframe ("MVS") datasets to and from remote sites in compliance with the SFTP or FTP protocol (SLiKSFTP operates as an SFTP or FTP client). MVS datasets are transferred directly to remote sites. There is no need to copy the data to the USS file system

SLiKSFTP also has built in compression routines that enable it to convert an MVS dataset to a zip file "on the fly". When this option is chosen the compression and the transfer operations proceed in parallel. Typically the transfer time will be significantly less than either the time required to transfer the MVS dataset uncompressed, or the time required to first zip the MVS dataset and then transfer the resulting zip file.

ASE makes it easy for you to trial SLiKSFTP anonymously so you can decide for yourself if it will help your business computing be more effective.

Installing SLiKSFTP

SLiKSFTP is installed by downloading the install XMIT file from the ASE website www.ase.com.au/sliksftp/ssdemo.htm

- Download SLIKS110.INSTALL.ZIP to your PC workstation.
- Extract file SLIKS110.INSTALL.XMIT.
- Transfer file SLIKS110.INSTALL.XMIT to a z/OS dataset named: *hlq*.SLIKS110.INSTALL.XMIT
- Use dataset attributes: RECFM=FB, LRECL=80 & a suitable BLKSIZE=value.
- Specify BINARY during the transfers to prevent any translation.

- Use the following JCL to RECEIVE the XMIT file:

```
//TSO      EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
   DELETE SLIKS110.INSTALL
   RECEIVE INDA('hlq.SLIKS110.INSTALL.XMIT')

   DELETE SLIKS110.SFTPSAMP
   RECEIVE INDA(SLIKS110.INSTALL(SFTPSAMP))

   DELETE SLIKS110.SFTPLOAD
   RECEIVE INDA(SLIKS110.INSTALL(SFTPLOAD))

/*
```

This will create the load module library *hlq*.SLIKS110.SFTPLOAD and the sample library hlq.SLIKS110.SFTPSAMP.

Product Activation

The copy of SLiKSFTP you download from the ASE website is a full-function demo version. Once installed you must obtain a key in order to use it. A key, good for the current day only, may be obtained anonymously from the SLiKSFTP web site at http://www.ase.com.au/sliksftp/ssdemo.htm. Alternatively, you may complete a Trial Agreement, fax it to ASE and we will email you a key for a 30 day extended trial.

The key is a 34-character string that must be supplied to the demo version of SLiKZiP in a control statement so that the product will function. The control statement will look like this:

```
demokey nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn
```

The string of n's represents the 34-character key. If the key is invalid or has expired, you'll get this message and SLiKZiP will terminate immediately:

```
ZIP0207I DEMOKEY value missing or invalid for today's date, terminating
```

Installing SLiKSFTP license key

When you take out a license for SLiKSFTP from ASE you will not have to supply DEMOKEY statements to use the software. Instead ASE will supply you with a 12-month key. The key can be installed in the product by running the job in member LICENSE of the sample library.

During the duration of your license term ASE will automatically supply a replacement 12-month key 45 days prior to the expiry of the current key that you are using. SLiKSFTP will automatically start warning users, via message ZIP0213W, when the current key is less than 31 days from expiring.

Using SLiKSFTP

The program is designed to be used in z/OS batch.

Sample JCL for running SLiKSFTP:

```
//SLIKZ    EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
..control statements go here..
//
See SAMPLIB member RUNSFTP
```

SSH Connection Authentication for SFTP

When running as a SFTP client the connection to the remote site is established via SSH. As SLiKSFTP runs as a batch job a non-interactive authentication method such as public key authentication is required.

Refer to the **Chapter 7 Getting ready to use OpenSSH** in the **IBM Ported Tools for z/OS User's Guide (SA22-7985)** for details on how to configure SSH authentication.

Running as an FTP client

By default SLiKSFTP runs as a SFTP client. The FTP control statement can be used to run as FTP client. The FTP statement must be the first statement in the control statement file or the first statement after the demokey statement.

SLiKSFTP Control Statements

Control statements, or commands, are read via ddname SYSIN. They have the format:

```
Verb operands
```

Some statements can only be used in either SFTP mode or FTP mode. Statements can be continued over multiple lines using the "trailing plus" syntax:

```
compopt select('s*.dat,t*.rpt'),+
        ascii,+
        lf
```

When a statement is to be continued, make the last non-blank character, or the last non-blank character prior to any "/*", a "+" (plus sign). If the following line has any leading spaces, they will be ignored. This means that if the statement syntax requires a blank or space, to separate two process options for example, then at least one space must precede the "+".

Comments can be added to the end of any control statement. Comments begin with the characters "/*" e.g:

```
Demokey C96A98378CC7DA37572DB107CE9F6AFC31 /* Today's key
```

The following commands are supported:

demokey *key*

Usage: SFTP and FTP mode

Specifies the trial license ***key.***
Key values, current for today only, may be obtained from the SLiKSFTP page at http://www.ase.com.au/sliksftp/ssdemo.htm

Trial users must provide a valid enabling key by placing a **demokey** statement as the first line in the control statement file.

[Licensed users of SLiKSFTP are provided with a long-term key which they install in the SLiKSFTP load module, thus removing the need for the demokey statement.]

ftp

Usage: FTP mode only

Switch to FTP mode. The FTP protocol will be used to transfer data to/from the server. The **ftp s**tatement should follow the **demokey** statement and precede all FTP commands.

open ***hostname [port]***

Usage: FTP mode only

Specifies the remote host to logon to. Optionally you can specify a port number to override the default port 21.

userid ***username@hostname***

Usage: SFTP mode only

Specifies the remote host to logon to and the user name to use. A session is established with the remote host via ssh.

userid ***username***

Usage: FTP mode only

Specifies the username to use when logging on to the remote ftp server.

pass ***password***

Usage: SFTP and FTP mode

Specifies the password for the userid specified on the userid statement.

cd ***remotedir***

Usage: SFTP and FTP mode

Change directory on the remote host. This command enables the user to identify where in the remote host's filesystem subsequent file & directory management commands operate.

binary

Usage: SFTP and FTP mode

Perform a binary transfer. The file being transferred is not to be subject to any form of translation.

ascii

Usage: SFTP and FTP mode

During a PUT or PUTZIP operation every byte of the data to be transferred will be first translated from EBCDIC to ASCII and LF sequences will be inserted as record terminators

During a GET or GETZIP operation the decrypted and unzipped data will be translated from ASCII to EBCDIC. LF or CRLF sequences will be removed from the end of each record.

crlf

Usage: SFTP and FTP mode

During a PUT or PUTZIP operation a carriage return and line feed sequence will be inserted at the end of each record before it is processed.

During a GET or GETZIP operation CRLF characters will be removed from the end of each record.

lf

Usage: SFTP and FTP mode

During a PUT or PUTZIP operation a line feed sequence will be inserted at the end of each record before it is processed.

During a GET or GETZIP operation line feed characters will be removed from the end of each record.

ebcdic

Usage: SFTP and FTP mode

During a PUT or PUTZIP a EBCDIC new line character (x'15') will be inserted at the end of each record.

During a GET or GETZIP operation the EBCDIC new line character (x'15) will mark the end of each logical record and will be removed.

The ebcdic option should be used when transferring text data between z/OS systems.

put ***mvsdsn|dd:ddname remotefile***

Usage: SFTP and FTP mode

Transfer a z/OS dataset to file ***remotefile*** on the remote host. The file is not compressed. Specify either a dataset name (***mvsdsn***) or a ddname to which the dataset is allocated (***dd:ddname***).

putzip ***mvsdsn|dd:ddname remotefile***

Usage: SFTP and FTP mode

Transfer z/OS dataset to file ***remotefile*** on the remote host. The z/OS dataset content will be compressed on the fly and ***remotefile*** will be in the zip file format. Specify either a dataset name (***mvsdsn***) or a ddname to which the dataset is allocated (***dd:ddname***).

If the ***mvsdsn*** is specified as * then the compression program will read control statements from the SLIKCNTL DD statement to determine the data to be compressed and the compression options. This feature allows more advanced zip files to be transferred to the remote system.

Get ***remotefile mvsdsn|dd:ddname***

Usage: SFTP and FTP mode

Transfer file ***remotefile*** from remote host to a z/OS dataset. Specify either a dataset name (***mvsdsn***) or a ddname to which the dataset is allocated (***dd:ddname***).

getzip ***mvsdsn|dd:ddname remotezipfile***

Usage: SFTP and FTP mode

Transfer file ***remotezipfile*** on the remote host, unzipping it to create a z/OS dataset. Specify either a dataset name (***mvsdsn***) or a ddname to which the dataset is allocated (***dd:ddname***).

If the ***mvsdsn*** is specified as * then the uncompression program will read control statements from the SLIKCNTL DD statement to determine the data to be uncompressed and the uncompression options. This feature allows more advanced uncompress operations to be performed.

chmod ***mode remotefile***

Usage: SFTP mode only

Changes the permissions of the file ***remotefile*** to ***mode***. Mode must be an octal number.

chgrp ***grp remotefile***

Usage: SFTP mode only

Changes the group the file ***remotefile*** to ***grp***. ***grp*** must be a numeric GID.

chown ***owner remotefile***

Usage: SFTP mode only

Changes the owner of file ***remotefile*** to ***owner***. ***owner*** must be a numeric UID.

exit|bye|quit

Usage: SFTP and FTP mode

Terminate the ssh session with the remote host.

ls ***remotefile***

Usage: SFTP and FTP mode

List statistics for file ***remotefile*** on the remote host. If remotefile is a directory the contents of the directory are listed. If directory is not specified the current working directory is listed.

mkdir ***dirname***

Usage: SFTP and FTP mode

Make directory ***dirname*** on the remote host.

rmdir ***dirname***

Usage: SFTP and FTP mode

Remove directory ***dirname*** on the remote host.

rm ***remotefile***

Usage: SFTP and FTP mode

Remove file ***remotefile*** from the "current" directory on the remote host.

pwd

Usage: SFTP and FTP mode

Print the name of the "working" or "current" directory on the remote host.

sshconf ***configfilename***

Usage: SFTP mode only

Specifies the name of an alternate ssh configuration file to be passed to ssh. This will override the system ssh options. This statement must occur before the USERID statement.

sshopt ***option***

Usage: SFTP mode only

Specifies an option to be passed to ssh. This allows individual options in the ssh config file to be overridden. This statement must occur before the USERID statement.

passive

Usage: FTP mode only

Toggles the state of the passive transfer mode. If passive mode is on then the server will listen on a data port and wait for a connection from the client rather than initiate a session on the receipt of a transfer command. By default passive mode in on.

Passive mode is necessary if the client is located behind a firewall. Normally the server will not be able to start a data session with the client if the session must pass through a firewall.

site ***siteoptions***

Usage: FTP mode only
The site command allows options that are unique to the server to be sent.

quote ***string***

Usage: FTP mode only
The free format string specified will be sent to the server as a command. This allows custom commands not implemented by SLiKSFTP to be sent to the server.

stat

Usage: FTP mode only
Displays a status response from the server.

syst

Usage: FTP mode only
Displays the type of operation system at the server.

help

Usage: FTP mode only
Displays the help information from the server.

options ***option1[,option2,...]***

Usage: SFTP and FTP mode

The OPTIONS statement or command supports the following keywords:

- **nostderr**

  Messages normally written to ddname STDERR will be discarded.

- **mergeout**

  Messages normally written to ddname STDERR will be written to STDOUT.

- **replace**

  A PUT or PUTZIP operation will replace an existing file of the same name on the remote host. If both REPLACE and NOREPLACE are omitted, REPLACE is the default.

- **noreplace**

  A PUT or PUTZIP operation will not replace an existing file of the same name on the remote host. If both RESUME and NORESUME are omitted, NORESUME is the default.

- **resume**

  A PUT or PUTZIP operation will resume a previously interrupted transfer of the same file to the remote host. If both RESUME and NORESUME are omitted, NORESUME is the default.

- **noresume**

  A PUT or PUTZIP operation will not resume a previously interrupted transfer of the same file to the remote host. The transfer will be restarted from the beginning of the file. If both RESUME and NORESUME are omitted, NORESUME is the default.

- **progress**

  Show progress messages during transfer. If both PROGRESS and NOPROGRESS are omitted, PROGRESS is the default.

- **noprogress**

  Do not show progress messages during transfer. If both PROGRESS and NOPROGRESS are omitted, PROGRESS is the default.

- **zipmsgs**

  Causes messages from the zip/unzip program to be written to the SYSPRINT DD statement. This will show compression statistics for PUTZIP and GETZIP operations. This is the default.

- **nozipmsgs**

  Suppresses the printing of messages from the zip/unzip program.

- **mow(nnnn)**

  Maximum outstanding writes. For each write the server responds with a status message. Performance is improved if these messages are processed in a group rather than individually. The mow(nnnn) parameter specifies the number of write requests that will be performed before a status response is read from the server. The default value is 1000. This parameter only affects SFTP transfers.

- **ignorerr**

  Causes errors reported by the FTP server to be ignored. This can be used to avoid SLiKSFTP terminating if a command such as RM fails due to a non existent file.

- **noignorerr**

  Non zero return codes from FTP commands will cause SLiKSFTP to terminate. This is the default.

Copyright Australian Systems Engineering Pty. Ltd. 2014

- **size(nnnn)**

  Sets the maximum packet size to use when transferring data. The maximum value is 98304. This value should be larger than the record length of the file being transferred.

- **keepalive(nnnn)**

  Sets the TCP_KeepAlive socket option to the value specified (1-2147460 seconds). The default is the system tcp value. This parameter allows you to override the system value if TCP sessions are timing out. TCP will send a keep-alive packet at the specified interval to keep the session active.

- **retry(nnnn)**

  Sets the maximum number of times that SLiKSFTP will retry a transfer that has failed due to a network error. The default is 5.

compopt ***option1[,option2,...]***

Usage: SFTP and FTP mode

This statement specifies options for the compress/uncompress process when performing a PUTZIP or GETZIP operation. The following keywords are supported:

- **ascii**

  During a PUTZIP operation every byte of the data to be transferred will be first translated from EBCDIC to ASCII.

  During a GETZIP operation the decrypted and unzipped data will be translated from ASCII to EBCDIC.

  Note that this differs from ASCII control statement above. The ASCII control statement also appends a carriage return and line feed to each record. This option just translates the record to ASCII, you then must specify either LF or CRLF to append line terminator characters.

- **binary**

  Perform a binary transfer. The file being transferred is not to be subject to any form of translation.

- **lf**

  During a PUTZIP each record transferred will have a linefeed character appended as a record terminator before being compressed.

During a GETZIP operation the end of each decrypted record will be determined by the presence of a linefeed character. The linefeed character will be removed before writing the record to the output dataset.

- **crlf**

  During a PUTZIP each record transferred will have a carriage return and linefeed characters appended as a record terminator before being compressed.

  During a GETZIP operation the end of each decrypted record will be determined by the presence of a carriage return/linefeed pair. The carriage return and linefeed characters will be removed before writing the record to the output dataset.

- **member(*name*)**

  Used when performing a PUTZIP operation to specify the member name in the zip file for the compressed mvs dataset.

- **opt(*string*)**

  Specifies a free format string of options to be passed to the compression program.

- **password(*password*)**

  For a PUTZIP operation, the zip file member is to be encrypted using AES 128/192/256 bit encryption using ***password***.

  For a GETZIP operation, if the zip file member is encrypted using AES 128/192/256 bit encryption, decryption will be attempted using ***password***.

  Note that ***password*** may be up to 99 characters in length. ***password*** must be enclosed in doublequotes (") if it contains spaces, commas, apostrophes or parentheses.

  Note that for a PUTZIP operation, ***password*** will be required at the remote system to unzip the file.

- **replace**

  During a GETZIP operation if the target MVS dataset is a PDS then existing members that are uncompressed will be replaced.

- **noreplace**

  During a GETZIP operation if the target MVS dataset is a PDS then existing members will are uncompressed will not be replaced. This is the default.

- **strip**

  Removes trailing blanks from each text record before transferring the record during a PUT or PUTZIP operation. Either ASCII or EBCDIC command must

also be specified. This will reduce the amount of data transferred especially if the input file has RECFM=FB. This is the default for text files.

- **nostrip**

  Suppresses removal of trailing blanks from each text record before transferring the record during a PUT or PUTZIP operation.

- **select(*selectionmask*)**

  For a PUTZIP operation, if the mvs dataset specifies a PDS then the selection mask is used to select members of the PDS to be added to the zip file. The selection mask has the form *member.ext* The *member* part is used to select the members of the PDS and can be generic e.g. D*.txt . The *ext* part is appended to each member name selected to create the default name for the zip file member.

  For a GETZIP operation, the selection mask is used to select members of the zip file to be unzipped. The selection mask can be generic e.g. Report*.dat.

- **text**

  During a PUTZIP operation every byte of the data to be transferred will be first translated from EBCDIC to ASCII and each record will have a carriage return and line feed characters appended.

  During a GETZIP operation the decrypted and unzipped data will be translated from ASCII to EBCDIC and carriage return and line feed characters will be removed from each record.

- **trunc**

  During a GETZIP operation, if the uncompressed record is longer than the logical record for the output dataset then the record will be truncated.

- **wrap**

  During a GETZIP operation, if the uncompressed record is longer than the logical record for the output dataset then the record will be wrapped into multiple records. This is the default.

- **zip64**

  Specifies that the 64-bit zip file architectural option is to be used. This option must be specified if there is a possibility that the z/OS dataset contains 4GB or more of data. Compliant unzip programs on the remote system will support the ZIP64 architecture.

/* (comment-only lines)

A comment-only line may be inserted at any point in the control statement file. Note that at least the first position of a comment-only line must be blank. The first, or left-most, non-blank characters on a comment-only line must be "/*".

Do not place "/*" in the first two positions of a control statement line as this will be interpreted by z/OS as an end-of-file indicator.

Sample Jobs

1. Transfer file, change file's mode on remote host:

```
//SLIKZ    EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
binary
put 'ASE.BACKUP.INCR01' backup.incr01
chmod 644 backup.incr01
exit
//
See SAMPLIB member PUTSFTP1
```

2. ZIP & transfer file, change file's mode on remote host:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
binary
putzip 'ASE.BACKUP.INCR01' backup.incr01.zip
chmod 644 backup.incr01.zip
exit
//
See SAMPLIB member ZIPSFTP1
```

3. ZIP, encrypt & transfer a file, set file's mode on remote host:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
binary
compopt password(thisismypassword)
putzip 'ASE.BACKUP.INCR01' backup.incr01.zip
chmod 644 backup.incr01.zip
exit
//
See SAMPLIB member ENCSFTP1
```

4. Translate file from EBCDIC to ASCII & transfer it:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
ascii
put 'ASE.DAILY.REPORT' report.txt
chmod 644 report.txt
exit
//
See SAMPLIB member PUTSFTP2
```

5. Zip a file, translating from EBCDIC to ASCII, and transfer it.

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
ascii
putzip 'ASE.DAILY.REPORT' report.zip
chmod 644 report.txt
exit
//
See SAMPLIB member ZIPSFTP2
```

Copyright Australian Systems Engineering Pty. Ltd. 2014

6. Retrieve a file from remote host.

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
binary
get database.backup 'ASE.DATABASE.BACKUP'
exit
//
See SAMPLIB member GETSFTP1
```

7. Retrieve a text file, translating from ASCII to EBCDIC:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
ascii
get newreport.txt 'ASE.NEW.REPORT'
exit
//
See SAMPLIB member GETSFTP2
```

8. Translate file from EBCDIC to ASCII without removing trailing blanks & transfer it:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
ascii
options nostrip
put 'ASE.DAILY.REPORT' report.txt
chmod 644 report.txt
exit
//
See SAMPLIB member PUTSFTP3
```

9. Transfer an EBCDIC text file to another EBCDIC machine:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ibm.com.au
cd /u/fred
ebcdic
put 'ASE.DAILY.REPORT' report.txt
chmod 644 report.txt
exit
//
See SAMPLIB member PUTSFTP4
```

Copyright Australian Systems Engineering Pty. Ltd. 2014

10. ZIP & transfer file, suppress zip messages:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
binary
options nozipmsgs
putzip 'ASE.BACKUP.INCR01' backup.incr01.zip
exit
//
See SAMPLIB member ZIPSFTP3
```

11. Transfer a ZIP file from a remote machine and UNZIP:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
binary
getzip backup.incr01.zip 'ASE.BACKUP.INCR01.NEW'
exit
//
See SAMPLIB member GETSFTP3
```

12. Transfer a ZIP file from a remote machine, UNZIP and decrypt:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
binary
compopt password(thisismypassword)
getzip backup.incr01.zip 'ASE.BACKUP.INCR01.NEW'
exit
//
See SAMPLIB member GETSFTP4
```

13. ZIP & transfer file using FTP protocol:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
ftp
open ase.com.au
userid fred
pass mypassword
cd zipfiles
binary
putzip 'ASE.BACKUP.INCR01' backup.incr01.zip
ls backup.incr01.zip
quit
//
See SAMPLIB member ZIPFTP1
```

Copyright Australian Systems Engineering Pty. Ltd. 2014

14. ZIP, encrypt & transfer a file using FTP protocol:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
ftp
open ase.com.au
userid fred
pass mypassword
binary
cd zipfiles
compopt password(thisismypassword)
putzip 'ASE.BACKUP.INCR01' backup.incr01.zip
quit
//
See SAMPLIB member ZIPFTP2
```

15. ZIP selected members from a PDS, translate to ascii, append linefeed to each record and transfer the zip file using FTP protocol:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
ftp
open 192.168.3.129
user peterm
pass password
compopt select('s*.dat,t*.rpt'),+
        ascii,+
        lf
putzip 'ASEPMD.ASE.CNTL' work.zip
ls work.zip
quit
//
See SAMPLIB member ZIPFTP3
```

16. Use SLiKZIP control statements to ZIP multiple datasets, translating to ascii , append carriage return/linefeed to each record, remove trailing blanks and transfer the zip file using FTP protocol:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
ftp
open 192.168.3.129
user peterm
pass password
putzip * work.zip
ls work.zip
quit
//SLIKCNTL DD *
zip OVERWRITE
fromds ase.ase.cntl
s s*.dat ascii lf
fromds asepmd.ase.cntl
s s*.jcl ascii lf
todd (.exit)
//
See SAMPLIB member ZIPFTP4
```

17. Transfer a ZIP file from a remote machine and UNZIP a selected member translating from ascii and removing the line feed from the end of each record:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
userid fred@ase.com.au
cd /home/fred
binary
compopt ascii,lf,select(ssh2.jcl),replace
getzip work.zip 'ASE.UNZIP.CNTL'
exit
//
See SAMPLIB member GETSFTP5
```

18. ZIP & transfer file using an alternate ssh configuration:

```
//SFTP     EXEC PGM=SLIKSFTP,REGION=4M
//STEPLIB  DD DISP=SHR,DSN=ASE.SLIKSFTP.LOAD
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSIN    DD *
Demokey 4EE5D584A133FEC2569394A6375364E48E
sshconf /u/fred/.ssh/new_config
userid fred@ase.com.au
cd /home/fred
binary
putzip 'ASE.BACKUP.INCR01' backup.incr01.zip
exit
//
See SAMPLIB member ZIPSFTP3
```

Copyright Australian Systems Engineering Pty. Ltd. 2014

Sample Output

1. The output of a typical SLIKSFTP job using SFTP protocol

```
SLIKSFTP 1.0 - Copyright ASE 2012
Mod level ASP1247 121012-1258
sftp> Demokey 72A3ACE12D17E895CF4E54B1BCAEB821CE

sftp> userid peterm@192.168.3.129
logging on to server
logon successful

sftp> cd /home/peterm
cd ok

sftp> binary
binary mode set

sftp> putzip 'ASE.BACKUP.INCR01' big2.zip
20121012-172313-FRI Transfered      1080308 bytes 00036010 b/s
20121012-172344-FRI Transfered      2119010 bytes 00034623 b/s
20121012-172404-FRI Transfered      3166549 bytes 00055133 b/s
Transfered      4076707 bytes in 000000087 secs 00046858 b/s
-rw-r--r--    1 peterm   users     4076707 Oct 12 17:25 big2.zip

sftp> chmod 644 big2.zip

sftp> exit
Child process terminated
SLIKSFTP terminating
OpenSSH_3.5p1, SSH protocols 1.5/2.0, OpenSSL 0x0090702f
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Rhosts Authentication disabled, originating port will not be trusted.
debug1: ssh_connect: needpriv 0
debug1: Connecting to 192.168.3.129 [192.168.3.129] port 22.
debug1: Connection established.
debug1: identity file /u/peterm/.ssh/id_rsa type 1
debug1: identity file /u/peterm/.ssh/id_dsa type -1
debug1: Remote protocol version 1.99, remote software version OpenSSH_4.1
debug1: match: OpenSSH_4.1 pat OpenSSH*
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_3.5p1
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client aes128-cbc hmac-md5 none
debug1: kex: client->server aes128-cbc hmac-md5 none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: dh_gen_key: priv key bits set: 124/256
debug1: bits set: 1022/2048
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
debug1: Host '192.168.3.129' is known and matches the RSA host key.
debug1: Found key in /u/peterm/.ssh/known_hosts:1
debug1: bits set: 1022/2048
debug1: ssh_rsa_verify: signature correct
debug1: kex_derive_keys
debug1: newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: waiting for SSH2_MSG_NEWKEYS
debug1: newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: done: ssh_kex2.
debug1: send SSH2_MSG_SERVICE_REQUEST
debug1: service_accept: ssh-userauth
```

```
debug1: got SSH2_MSG_SERVICE_ACCEPT
debug1: authentications that can continue: publickey,keyboard-interactive
debug1: next auth method to try is publickey
debug1: try pubkey: /u/peterm/.ssh/id_rsa
debug1: input_userauth_pk_ok: pkalg ssh-rsa blen 149 lastkey 1203e0a8 hint 0
debug1: read PEM private key done: type RSA
debug1: ssh-userauth2 successful: method publickey
debug1: fd 7 setting O_NONBLOCK
debug1: fd 8 setting O_NONBLOCK
debug1: fd 9 setting O_NONBLOCK
debug1: channel 0: new [client-session]
debug1: send channel open 0
debug1: Entering interactive session.
debug1: ssh_session2_setup: id 0
debug1: Sending subsystem: sftp
debug1: channel request 0: subsystem
debug1: channel 0: open confirm rwindow 0 rmax 32768
SLIKZIP 2.0  - Copyright ASE  2012                          Serial No:Demo****
ZIP0007I levels: P1225 121012-0948,P1225 121012-0948,P1225 121012-0948
ZIP0087I PARM= demokey(72A3ACE12D17E895CF4E54B1BCAEB821CE) CTLDD() ZIP FROMDD(+
 SYS00001) TODD(.EXIT) OVERWRITE

ZIP0149I Control statement listing...

<---+----1----+----2----+----3--statement area---5----+----6----+----7----+---->
ZIP0243I exits enabled: OPEN,CLOS,WRIT
ZIP0213W WARNING: License key expires today
-------------------------------------------------------------------------------
ZIP0091I fri 17:22:45 start ZIP  todd(.EXIT)
ZIP0103I reading PS U   dataset  : ASE.BACKUP.INCR01
ZIP0178I options are: ADD
Member   Ext_used    I/p_size   Strips_to    O/p_size Save Path/filename
                     40545344    40545344     4076559 90%  ASE.BACKUP.INCR01
process totals:      40545344    40545344     4076559 90%      1 members
overall totals:      40545344    40545344     4076559 90%      1 members
ZIP0083I fri 17:24:12 end   ZIP      returned rc 0000, cpu:    69.4 sec
-------------------------------------------------------------------------------
ZIP0206I terminating with COND CODE 0000
```

2. The output of a typical SLIKSFTP job using FTP protocol

```
SLIKSFTP 1.0 - Copyright ASE 2013
Mod level ASP1284 130118-1842
sftp> Demokey B405CDD5688B363DF3E1DD18E86AB8A7A3

sftp> ftp
ftp mode set

sftp> open haystak.ase.com.au
220 "Welcome to FTP service."

sftp> user peterm
331 Please specify the password.

sftp> ** password suppressed **
230 Login successful.

sftp> binary
binary mode set

sftp> putzip 'ASE.DAILY.REPORT.D070802.T16' d070802.t16
227 Entering Passive Mode (192,168,3,129,95,69)
200 Switching to Binary mode.
150 Ok to send data.
20130121-114830-MON Transfered      1052724 bytes 00040489 b/s
20130121-114853-MON Transfered      2097976 bytes 00047511 b/s
20130121-114921-MON Transfered      3177880 bytes 00038568 b/s
```

```
20130121-114937-MON Transfered      4206286 bytes 00068560 b/s
20130121-115002-MON Transfered      5253675 bytes 00041895 b/s
20130121-115011-MON Transfered      6315918 bytes 00118027 b/s
20130121-115016-MON Transfered      7360840 bytes 00208984 b/s
20130121-115025-MON Transfered      8390173 bytes 00128666 b/s
20130121-115033-MON Transfered      9448482 bytes 00132288 b/s
20130121-115038-MON Transfered     10502579 bytes 00210819 b/s
20130121-115048-MON Transfered     11564536 bytes 00117995 b/s
20130121-115052-MON Transfered     12596688 bytes 00258038 b/s
20130121-115100-MON Transfered     13660597 bytes 00132988 b/s
20130121-115109-MON Transfered     14680137 bytes 00127442 b/s
20130121-115117-MON Transfered     15757138 bytes 00134625 b/s
20130121-115121-MON Transfered     16798132 bytes 00260248 b/s
20130121-115131-MON Transfered     17827434 bytes 00114366 b/s
20130121-115135-MON Transfered     18894351 bytes 00266729 b/s
20130121-115144-MON Transfered     19935179 bytes 00115647 b/s
20130121-115149-MON Transfered     20990781 bytes 00263900 b/s
20130121-115157-MON Transfered     22026819 bytes 00129504 b/s
20130121-115200-MON Transfered     23089149 bytes 00354110 b/s
20130121-115209-MON Transfered     24134299 bytes 00130643 b/s
20130121-115213-MON Transfered     25182256 bytes 00261989 b/s
20130121-115221-MON Transfered     26217916 bytes 00129457 b/s
20130121-115224-MON Transfered     27273057 bytes 00351713 b/s
20130121-115231-MON Transfered     28330964 bytes 00176317 b/s
20130121-115234-MON Transfered     29383508 bytes 00350848 b/s
20130121-115238-MON Transfered     30432018 bytes 00262127 b/s
20130121-115244-MON Transfered     31483741 bytes 00175287 b/s
20130121-115247-MON Transfered     32532251 bytes 00349503 b/s
20130121-115250-MON Transfered     33580888 bytes 00349545 b/s
20130121-115258-MON Transfered     34604819 bytes 00146275 b/s
20130121-115302-MON Transfered     35653998 bytes 00262294 b/s
20130121-115310-MON Transfered     36728835 bytes 00134354 b/s
20130121-115319-MON Transfered     37772411 bytes 00130447 b/s
20130121-115324-MON Transfered     38808099 bytes 00207137 b/s
20130121-115332-MON Transfered     39853610 bytes 00130688 b/s
20130121-115341-MON Transfered     40922098 bytes 00133561 b/s
20130121-115345-MON Transfered     41963063 bytes 00260241 b/s
20130121-115355-MON Transfered     43023576 bytes 00117834 b/s
20130121-115359-MON Transfered     44068662 bytes 00261271 b/s
Transfered     44415223 bytes in 000000354 secs 00125466 b/s
226 File receive OK.

sftp> ls d070802.t16
227 Entering Passive Mode (192,168,3,129,178,8)
150 Here comes the directory listing.
-rw-r--r--    1 1000     100      44415223 Jan 21 00:57 d070802.t16
226 Directory send OK.
sftp> quit
221 Goodbye.
SLIKSFTP terminating
SLIKZIP 2.0  - Copyright ASE  2012                          Serial No:Demo****
ZIP0007I levels: P1225 121109-1319,P1225 121109-1319,P1225 121109-1319
ZIP0087I PARM= demokey(B405CDD5688B363DF3E1DD18E86AB8A7A3) CTLDD() ZIP FROMDD(+
 SYS00008) TODD(.EXIT) OVERWRITE

ZIP0149I Control statement listing...

<---+----1----+----2----+----3--statement area---5----+----6----+----7----+---->
ZIP0243I exits enabled: PUMS,OPEN,CLOS,WRIT
ZIP0213W WARNING: License key expires today
-------------------------------------------------------------------------------
ZIP0091I mon 11:48:07 start ZIP  todd(.EXIT)
ZIP0103I reading PS U   dataset  : ASE.DAILY.REPORT.D070802.T16
ZIP0178I options are: ADD
Member   Ext_used    I/p_size   Strips_to    O/p_size Save Path/filename
                    148483328   148483328    44415049 70%  ASE.DAILY.REPORT.D070802.T16
process totals:     148483328   148483328    44415049 70%      1 members
overall totals:     148483328   148483328    44415049 70%      1 members
ZIP0083I mon 11:54:01 end   ZIP      returned rc 0000, cpu:   320.2 sec
-------------------------------------------------------------------------------
ZIP0206I terminating with COND CODE 0000
```

Messages

ZIP601E Missing Continuation Card

The end of the control statement file was reached before the expected continuation statement for the last statement was found. The last statement ended with a + character which indicates that the statement is continued on the next line.

ZIP602I Child process terminated

The UNIX System Services task attached by SLiKSFTP has ended.

ZIP603I SLiKSFTP terminating rc=

SLiKSFTP is ending with the specified return code.

ZIP606E demokey statement required

A demokey statement must appear be the first statement in the control statement file.

ZIP604E unknown control statement

The control statement is not valid. Check the statement for syntax errors.

ZIP605E user command failed

The command failed. Check any error messages, correct problem and rerun the job.

ZIP607E cd command failed

The cd command failed. Check any error messages, correct problem and rerun the job.

ZIP608I binary mode set

All data will be transferred as binary characters. No translation will be performed.

ZIP609I ascii mode set

All data will be translated to/from ASCII.

ZIP610I ebcdic mode set

Copyright Australian Systems Engineering Pty. Ltd. 2014

All data will be translated to/from EBCDIC. All x'15' characters will be treated as end of record characters.

ZIP611E put command failed

The put command failed. Check any error messages, correct problem and rerun the job.

ZIP612E get command failed

The get command failed. Check any error messages, correct problem and rerun the job.

ZIP613I debug mode set

Diagnostic message will be issued.

ZIP614E mkd command failed

The mkd command failed. Check any error messages, correct problem and rerun the job.

ZIP615E rmd command failed

The rmd command failed. Check any error messages, correct problem and rerun the job.

ZIP616E rm command failed

The rm command failed. Check any error messages, correct problem and rerun the job.

ZIP617I ftp mode set

SLiKSFTP will connect to the remote server as an ftp client.

ZIP618I passive mode off

SLiKSFTP will not use passive mode to perform transfers. The server will open a session with the client to transfer data. This may not work if the client is behind a firewall. In this case passive mode should be used.

ZIP619I passive mode on

SLiKSFTP will use passive mode to perform all transfers. In passive mode the client will initiate a data session with the server.

ZIP620E site command failed

The site command failed. Check any error messages, correct problem and rerun the job.

ZIP621E quote command failed

The quote command failed. Check any error messages, correct problem and rerun the job.

ZIP622E command failed

The command failed. Check any error messages, correct problem and rerun the job.

ZIP623E sshconf must be before userid statement

The sshconf statement is misplaced. It must be before the userid statement.

ZIP624E sshopt must be before userid statement

The sshopt statement is misplaced. It must be before the userid statement.

ZIP625E pwd command failed

The pwd command failed. Check any error messages, correct problem and rerun the job.

ZIP625E ftp mode not set

SLiKSFTP is running in SFTP mode. This control statement can only be used in FTP mode.

ZIP626E sftp command, not supported in ftp mode

SLiKSFTP is running in FTP mode. This control statement can only be used in SFTP mode.

ZIP627E Child process terminated unexpectedly

The SLiKSFTP USS process has ended unexpectedly.

ZIP628E select failed

The TCP select command failed. Check error messages.

ZIP629E timeout

The TCP select command timed out. The connection to the server has failed.

ZIP633E read failed

TCP read command failed.

ZIP635E message buffer too small

The buffer to read messages into is too small. Contact ASE support

ZIP636E Unexpected message

SLiKSFTP received an unexpected message from the server. The message will be printed. Contact ASE support.

ZIP637E TO pipe creation failed

Creation of a UNIX pipe failed. Contact ASE for support.

ZIP638E FROM pipe creation failed

Creation of a UNIX pipe failed. Contact ASE for support.

ZIP639E stderr pipe creation failed

Creation of a UNIX pipe failed. Contact ASE for support.

ZIP640E fcntl getfl failed

The fcntl command failed. Contact ASE for support.

ZIP641E fcntl non blocking failed

The fcntl to make pipe non blocking failed. Contact ASE for support.

ZIP642E fork failed

Fork of child process failed. Contact ASE for support.

ZIP0643I SSH failed Ret=xxxx res=xxxx

The call to SSH has failed. This message lists the return and reason codes for the failure

ZIP0643I FTP failed Ret=xxxx res=xxxx

The call to FTP has failed. This message lists the return and reason codes for the failure

ZIP0644I child exit

SLiKSFTP USS process has ended.

ZIP0645I logging on to server

SLiKSFTP is attempting to logon to the remote server.

ZIP0646I attach failed for wait for child subtask

The wait for child subtask could not be attached. Contact ASE for support.

ZIP0648I logon successful

Logon to the remote server was successful

ZIP0649I logon failed

Logon to the remote server failed. Check the correct logon id has been specified and correct ssh keys have been setup.

ZIP648E allocation failed

Allocation for the mvs dataset failed. Check the correct dataset has n=been specified.

ZIP644E Named pipe creation failed

Creation of a named pipe failed.

ZIP645E Open failed for named pipe

The named pipe could not be opened.

ZIP646E Error during compress operation

An error occurred during the compress or uncompresses operation. Check the messages from SLIKZIP for more details.

ZIP647E Non zero return code during uncompress operation - check listing

An error occurred during the compress or uncompresses operation. Check the messages from SLIKZIP for more details.

ZIP648E allocation failed for dsname

The specified dataset could not be allocated. Check that the correct name has been specified.

ZIP649I Checking for incomplete previous transfer..

SLiKSFTP will check on the server for a previous transfer that did not complete.

ZIP650E size command failed

SLiKSFTP sent a SIZE command to the remote server. The command failed.

ZIP651I No incomplete previous transfer

There was no incomplete previous transfer

ZIP652I Incomplete file found

A previous transfer failed. SLiKSFTP has found an incomplete file and will resume the transfer.

ZIP653I checking file size is correct

The size of the file on the remote server wil be compared

ZIP654E SIZE command failed

The ftp SIZE command failed. The size of the file on the remote server cannot be determined.

ZIP655E Error during transfer - size is wrong, local xxxx remote xxxx

The size of the file on the remote server differs from the number of bytes transferred. The transfer has failed.

ZIP656I File size is correct

The size of the file on the remote server agrees with the number of bytes transferred.

ZIP657I Deleting existing file..

The existing file on the remote server will be deleted prior to the newly transferred file being renamed.

ZIP658E dele command failed

The delete command failed. SLiKSFTP was unable to delete the existing file. Check that you have permission to delete the file.

ZIP659I Renaming .filepart to correct name..

The temporary .filepart name is being renamed to the correct file name.

ZIP660E rnfm command failed

The ftp rename from command failed. The file cannot be renamed.

ZIP661E rnto command failed

The ftp rename to command failed. The file cannot be renamed. Check you have permission to rename the file

ZIP662I Transfer complete

The file transfer has completed successfully

ZIP664E File exists and noreplace specified, transfer aborted

The file to be transfered already exists on the remote server and the replace option has not be specified. The file will not be transferred.

ZIP665E open failed for mvs dataset

The mvs dataset could not be opened. Check the name is correct and that you have permission to open the dataset.

ZIP666E Packet size value too small

The value specified for the size option is too small. It needs to be at least two bytes larger than the record length of the file being transferred

ZIP667I Existing filepart found, xxxxxxxxxxx bytes, resuming transfer

An incomplete transfer file was found on the remote server. The transfer will resume from the specified point.

ZIP670I Transfered xxxxxxxxxxxx bytes in xxxxxxxxx secs xxxxxxxx b/s

This message lists the number of bytes transferred, then time taken and the bytes per second.

ZIP674E attach failed for TESTSLIK

The TESTSLIK program could not be attached. Contact ASE for support

ZIP675E attach failed for SLiKZiP

SLIKZIP could not be attached. Contact ASE for support.

ZIP676E No such file

The file does not exist on the remote system

ZIP680E Chmod failed

The chmod command failed. Check that you have permission to perform the operation

ZIP682E directory already exists

You attempted a mkdir command for a directory that already exists on the remote system.

ZIP683E mkdir failed

The mkdir command failed. Check that you have permission to perform the operation

ZIP685E rm failed

The mkdir command failed. Check that you have permission to perform the operation

ZIP687I cd ok

The cd command was completed successfully

ZIP688E not a directory

The cd command cannot be completed as the specified file is not a directory

ZIP689E read failed

Sliksftp was unable to read from a named pipe failed.

ZIP690E unexpected return from select

A TCP/IP select command ended with an unexpected return code.

ZIP691E socket write with translation failed

A write to a TCP/IP socket failed

ZIP692E Bad PASV|SIZE|SYST response

The response to the PASV command was not in the expected format.

ZIP694E Error during socket write

An error occurred while writing to a socket

ZIP697E Unexpected message or message format

The message was not expected or the format of the message differed from the expected format

ZIP698E pasv command failed

The PASV command did not return the response expected.

ZIP699I Restarting transfer at offset xxxxxxxxxxxxxx

An incomplete file transfer was found on the remote server. The transfer will be resumed at the specified offset.

ZIP707I YYYYMMDD-HHMMSS-DAY Transfered xxxxxxxxxx bytes xxxxxxxxx b/s

This is a transfer progress message. It details the number of bytes transferred.

ZIP708E setting setsockopt tcp_keepalive failed

The program was unable to set the TCP KEEPALIVE socket option.

ZIP709E setting setsockopt so_keepalive failed

The program was unable to set the TCP SO_KEEPALIVE socket option.

ZIP719E Invalid Demokey value

The demokey value is incorrect. Check that you have coded the correct value.

ZIP721E signal failed

An attempt to send a USS signal failed.

ZIP722I testslik

This message indicates that TESTSLIK is running

ZIP723I testslik ended

This message indicates that TESTSLIK has ended

ZIP724I Response:

This message documents the response to the previous command sent to the server.

ZIP725I SIZE command not supported - restart disabled

The SIZE command is not supported by the FTP server. The restart facility cannot be used with this server.

ZIP726I No incomplete previous transfer

SLiKSFTP did not find any incomplete transfer on the server. SLiKSFTP will transfer the complete file.

ZIP727E Unable to get attributes of remote file filename

SLiKSFTP was unable to read attributes for the specified file.

ZIP728E rename failed : oldname to newname

SLiKSFTP was unable to rename oldname to newname.

ZIP729E delete failed for filename

SLiKSFTP was unable to delete the specified file. Check that the user is authorized to delete the file.

ZIP731E Open failed for filename

The specified file could not be opened. Check that the user is authorized to open the file.

ZIP732E Transfer failed for filename

The transfer of the specified file failed. Check preceding messages for the cause of the failure.

ZIP733I CR LF will be appended to ASCII records

Each record translated to ASCII will have a carriage return and line feed appended.

ZIP734I renaming oldname to newname

File oldname will be renamed to file newname.

ZIP735I logon failed

Logon to the remote server failed. Check the correct logon id has been specified and correct ssh keys have been setup.

ZIP736I Socket error retc=xxxxxxxx reason=xxxxxxxx

An error occurred during a socket call. The return code and reason code are listed.

ZIP737I Wrong file offset

This is an internal logic error. Please report to ASE support.

ZIP738I Repositioning File

During a get operation, the zip file on the remote machine needs to be reopened and positioned to an earlier position in the file.

ZIP743I Connection error – restarting connection

An error has occurred with the connection to the remote machine causing the current transfer to fail. An attempt will be made to restart the connection and complete the transfer.

ZIP744I Connection error – Maximim retries reached, aborting

An error has occurred with the connection to the remote machine causing the current transfer to fail. The connection will not be restarted because the maximum number of retries has been reached.

Publication number: SS12-1012-6

This manual serves as a reference source for users of the SLiKSFTP product. Please use this form to comment on the publication in terms of its organization or content with the understanding that ASE may use or distribute the information that you supply in any way that it believes appropriate without incurring an obligation to you.

If you require further copies of this publication please order them through your ASE representative. If you require assistance in the installation, configuration or use of the SLiKSFTP product please contact ASE for support.

Suggested aspects that might be addressed by your comments include:

Clarity Accuracy Completeness Organization Examples Legibility

Please give your name and address if you require a reply:

name:

email:

Address:

Occupation:

Comments: